# Predicción de diagnóstico de Diabetes con ML

**Andrés Tejeda**
**23002674**


# Fuente de datos

Dataset del National Institute of Diabetes and Digestive and Kidney Diseases de Estados Unidos.

Limitante: todos los registros son de mujeres de al menos 21 años de edad de herencia Nativo-Americana.

URL: https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset

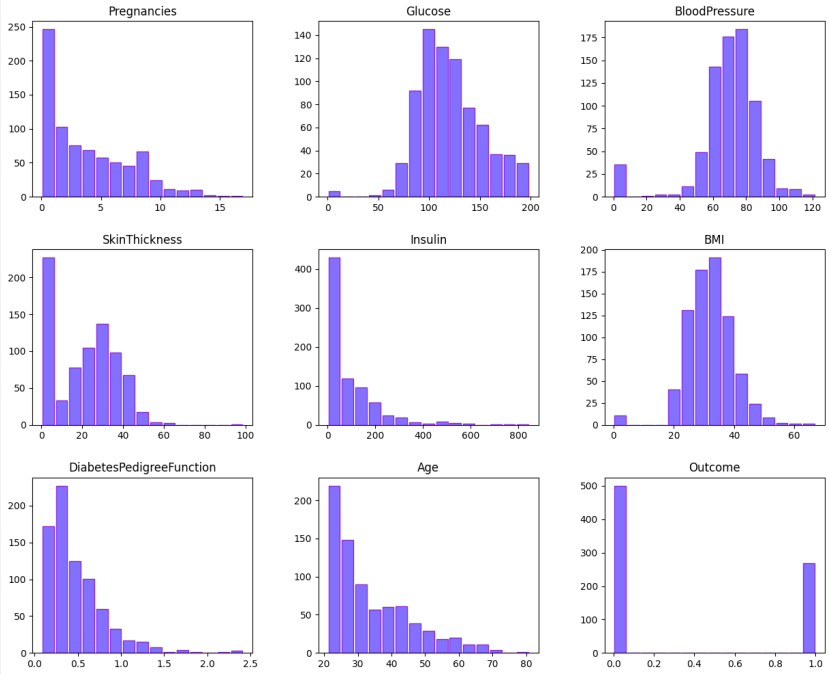

# Diccionario

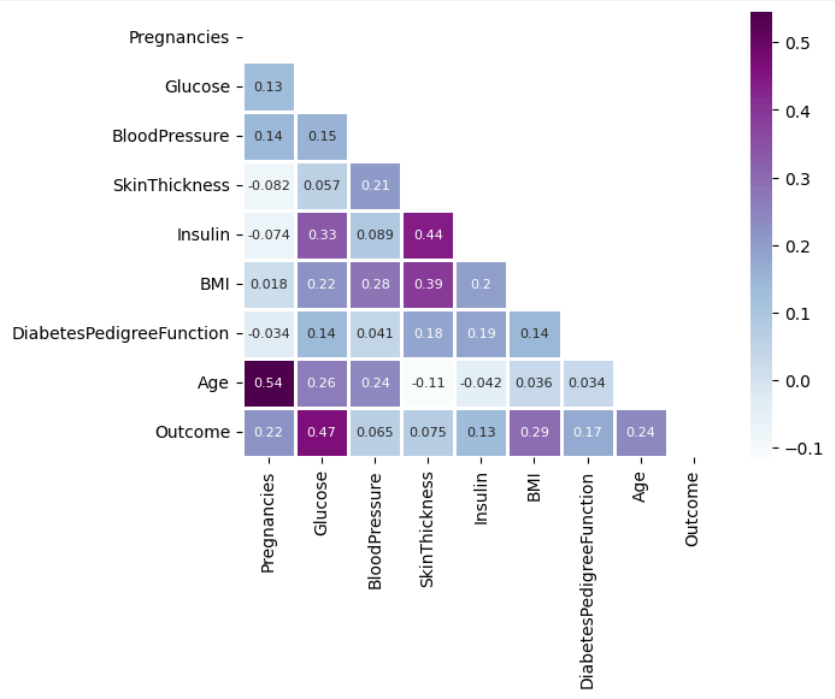
- **Pregnancies**: Cantidad de embarazos por persona
- **Glucose**: Nivel de glucosa en la sangre
- **BloodPressure**: Medición de presión sanguínea
- **SkinThickness**: Grosor de la pie
- **Insulin**: nivel de insulina en la sangre
- **BMI**: Índice de masa corporal
- **DiabetesPedigreeFunction**: Probabilidad de padecer diabeter por historial familiar
- **Age**: Edad
- **Outcome**: Resultado. 1: tiene diabetes. 0: no tiene diabetes

# Análisis y preparación de datos

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| count | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 |
| mean | 3.845052 | 120.894531 | 69.105469 | 20.536458 | 79.799479 | 31.992578 | 0.471876 | 33.240885 | 0.348958 |
| std | 3.369578 | 31.972618 | 19.355807 | 15.952218 | 115.244002 | 7.884160 | 0.331329 | 11.760232 | 0.476951 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.078000 | 21.000000 | 0.000000 |
| 25% | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | 0.243750 | 24.000000 | 0.000000 |
| 50% | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 30.500000 | 32.000000 | 0.372500 | 29.000000 | 0.000000 |
| 75% | 6.000000 | 140.250000 | 80.000000 | 32.000000 | 127.250000 | 36.600000 | 0.626250 | 41.000000 | 1.000000 |
| max | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | 2.420000 | 81.000000 | 1.000000 |

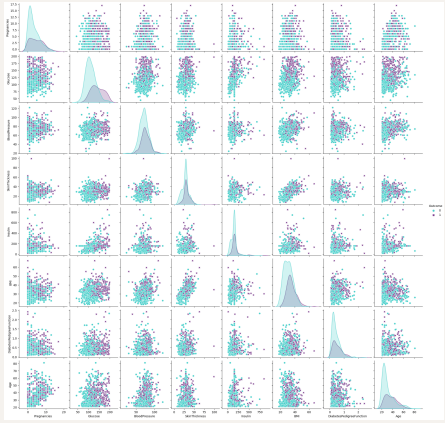
```
[190]  diabetes.isna().sum()

...    Pregnancies                 0
       Glucose                     0
       BloodPressure               0
       SkinThickness               0
       Insulin                     0
       BMI                         0
       DiabetesPedigreeFunction    0
       Age                         0
       Outcome                     0
       dtype: int64
```


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
0.13
0.14 0.15
-0.082 0.057 0.21
-0.074 0.33 0.089 0.44
0.018 0.22 0.28 0.39 0.2
-0.034 0.14 0.041 0.18 0.19 0.14
0.54 0.26 0.24 -0.11 -0.042 0.036 0.034
0.22 0.47 0.065 0.075 0.13 0.29 0.17 0.24
0.5
0.4
0.3
0.2
0.1
0.0
−0.1

## Original

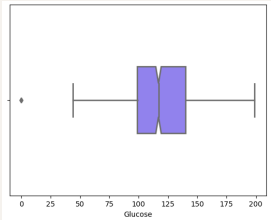
## Imputación por media

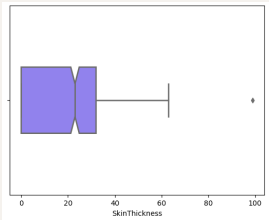


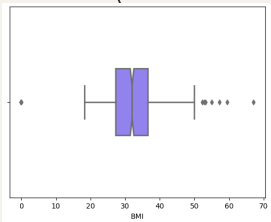
## Imputación por KNN

# Limpieza de datos

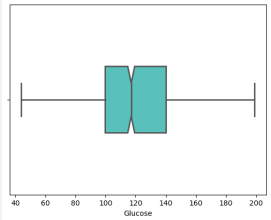
Original

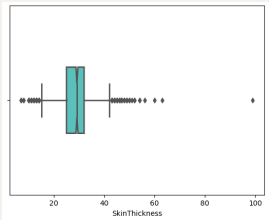


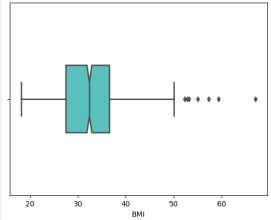



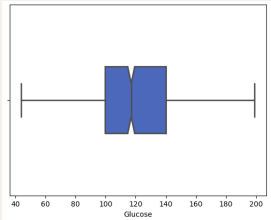

Imputación por media

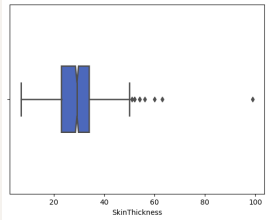


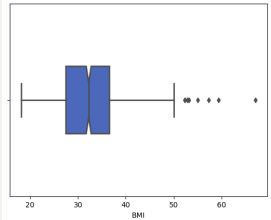



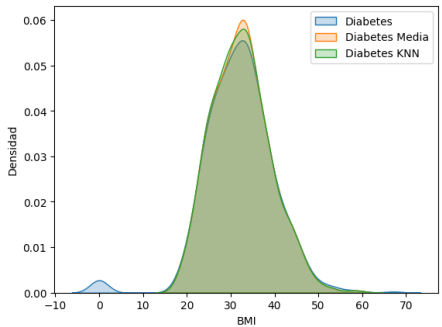



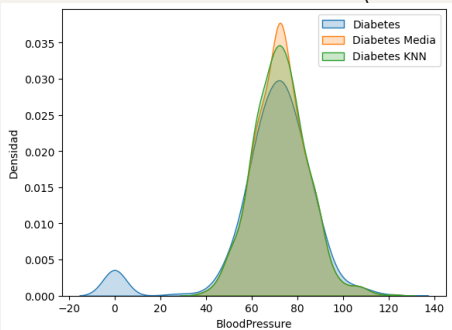

Imputación por KNN

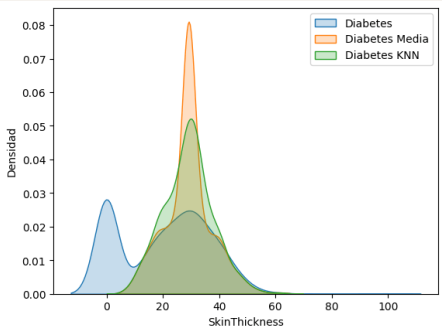


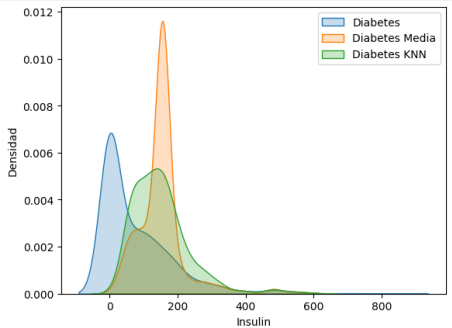




Diabetes
Diabetes Media
Diabetes KNN
Densidad
BMI


Diabetes
Diabetes Media
Diabetes KNN
Densidad
BloodPressure


Diabetes
Diabetes Media
Diabetes KNN
Densidad
SkinThickness

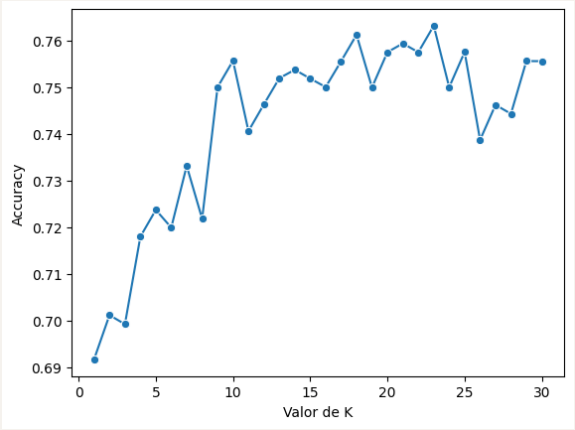

Diabetes
Diabetes Media
Diabetes KNN
Densidad
Insulin

# Modelo

```
seed = 42

# diabetes
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = seed)

# diabetes media imputada
xmed_train, xmed_test, ymed_train, ymed_test = train_test_split(x_med, y_med, test_size = 0.3, random_state = seed)

# diabetes KNN imputado
xknn_train, xknn_test, yknn_train, yknn_test = train_test_split(x_KNN, y_KNN, test_size = 0.3, random_state = seed)
```

```
sx_train = estandar.fit_transform(x_train)
sx_test = estandar.transform(x_test)

sxmed_train = estandar.fit_transform(xmed_train)
sxmed_test = estandar.transform(xmed_test)

sxknn_train = estandar.fit_transform(xknn_train)
sxknn_test = estandar.transform(xknn_test)
```

```python
rango2 = [i for i in range (1,31)]
acc2 = []

for k in rango2:
    knn = KNeighborsClassifier(n_neighbors=k)
    score = cross_val_score(knn, sxmed_train, ymed_train, cv=5)
    acc2.append(np.mean(score))
```



```
knn = KNeighborsClassifier(n_neighbors=mejor_k)
knn.fit(sx_train, y_train)

knn_med = KNeighborsClassifier(n_neighbors=mejor_kmed)
knn_med.fit(sxmed_train, ymed_train)

knn_knn = KNeighborsClassifier(n_neighbors=mejor_kknn)
knn_knn.fit(sxknn_train, yknn_train)
```

```
arbol = DecisionTreeClassifier(criterion="entropy", max_depth= 3)
arbol_med = DecisionTreeClassifier(criterion="entropy", max_depth = 3)
arbol_knn = DecisionTreeClassifier(criterion="entropy", max_depth = 3)
```

0.0s

```
arbol.fit(sx_train, y_train)
arbol_med.fit(sxmed_train, ymed_train)
arbol_knn.fit(sxknn_train, yknn_train)
```

```
y_pred = knn.predict(sx_test)
ymed_pred = knn_med.predict(sxmed_test)
yknn_pred = knn_knn.predict(sxknn_test)
```

```
arbol_pred = arbol.predict(sx_test)
arbol_predmed = arbol_med.predict(sxmed_test)
arbol_predknn = arbol_knn.predict(sxknn_test)
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.81 | 0.82 | 156 |
| 1 | 0.60 | 0.61 | 0.61 | 72 |
| accuracy | | | 0.75 | 228 |
| macro avg | 0.71 | 0.71 | 0.71 | 228 |
| weighted avg | 0.75 | 0.75 | 0.75 | 228 |

KNN con imputación por media

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.84 | 0.83 | 156 |
| 1 | 0.63 | 0.58 | 0.60 | 72 |
| accuracy | | | 0.76 | 228 |
| macro avg | 0.72 | 0.71 | 0.72 | 228 |
| weighted avg | 0.75 | 0.76 | 0.76 | 228 |

Árbol con imputación por media

# Insights

- Existen muchos registros con valores de 0
- No hay una correlación lineal clara
- Diferente acercamiento